

```
import pandas as pd
pd.set_option('display.max_columns', 10)
print("Original data set from Excel file:\n")
df=pd.read_excel("E:/hari/employees.xlsx")
print(df)
f=df['First Name'].str.strip()
f.to_excel("E:/hari/employees.xlsx")
print("\nReplace value with Forward:\n")
```

Original data set from Excel file:

```
-----
FileNotFoundError                                Traceback (most recent call last)
<ipython-input-1-94d6cdc3a4f1> in <cell line: 4>()
      2 pd.set_option('display.max_columns', 10)
      3 print("Original data set from Excel file:\n")
----> 4 df=pd.read_excel("E:/hari/employees.xlsx")
      5 print(df)
      6 f=df['First Name'].str.strip()

-----
3 frames
/usr/local/lib/python3.10/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mode, encoding, compression, memory_map,
is_text, errors, storage_options)
    880     else:
    881         # Binary mode
--> 882         handle = open(handle, ioargs.mode)
    883         handles.append(handle)
    884

FileNotFoundError: [Errno 2] No such file or directory: 'E:/hari/employees.xlsx'
```

Next steps: [Explain error](#)

```
k=df.fillna(method='pad')
print(k)
print("\nThe above updated data set will be stored in sample2.xlsx file...\n")
k.to_excel("E:/hari/employees.xlsx")
```

First Name Gender Start Date Last Login Time Salary Bonus % \

0	Douglas	Male	1993-08-06	12:42:00	97308.0	6.945
1	Thomas	Male	1996-03-31	06:53:00	61933.0	4.170
2	Maria	Female	1993-04-23	11:17:00	130590.0	11.858
3	Jerry	Male	2005-03-04	13:00:00	138705.0	9.340
4	Larry	Male	1998-01-24	16:47:00	101004.0	1.389
..
995	Henry	Male	2014-11-23	06:09:00	132483.0	16.655
996	Phillip	Male	1984-01-31	06:30:00	42392.0	19.675
997	Russell	Male	2013-05-20	12:39:00	96914.0	1.421
998	Larry	Male	2013-04-20	16:45:00	60500.0	11.985
999	Albert	Male	2012-05-15	18:24:00	129949.0	10.169

Senior Management Team

0	1.0	Marketing
1	1.0	Marketing
2	0.0	Finance
3	1.0	Finance
4	1.0	Client Services
..
995	0.0	Distribution
996	0.0	Finance
997	0.0	Product
998	0.0	Business Development
999	1.0	Sales

[1000 rows x 8 columns]

The above updated data set will be stored in sample2.xlsx file...

```
print("\nDrop one particular column and its values:\n")
k.drop(['Gender'], axis=1, inplace=True)
print(k)
print("The above updated data set will be stored in sample3.alsa file\n")
```

Drop one particular column and its values:

First Name Start Date Last Login Time Salary Bonus % \

0	Douglas	1993-08-06	12:42:00	97308.0	6.945
1	Thomas	1996-03-31	06:53:00	61933.0	4.170
2	Maria	1993-04-23	11:17:00	130590.0	11.858
3	Jerry	2005-03-04	13:00:00	138705.0	9.340
4	Larry	1998-01-24	16:47:00	101004.0	1.389

```

..      ...      ...
995      Henry 2014-11-23      06:09:00      132483.0      16.655
996      Phillip 1984-01-31      06:30:00      42392.0      19.675
997      Russell 2013-05-20      12:39:00      96914.0      1.421
998      Larry 2013-04-20      16:45:00      60500.0      11.985
999      Albert 2012-05-15      18:24:00      129949.0      10.169

```

```

Senior Management      Team
0      1.0      Marketing
1      1.0      Marketing
2      0.0      Finance
3      1.0      Finance
4      1.0      Client Services
..      ...      ...
995      0.0      Distribution
996      0.0      Finance
997      0.0      Product
998      0.0      Business Development
999      1.0      Sales

```

[1000 rows x 7 columns]

The above updated data set will be stored in sample3.alsa file

```

k.to_excel("E:/hari/employees.xlsx")
print("\nDrop NaN rows:\n")
df=pd.read_excel("E:/hari/employees.xlsx")
print(df)
x=df.dropna()
print(x)
print("\nThe above updated data set will be stored in sample5.xlsx file....\n")
x.to_excel("E:/hari/employees.xlsx")

```



Drop NaN rows:

```

Unnamed: 0 First Name Start Date Last Login Time Salary Bonus % \
0      0      Douglas 1993-08-06      12:42:00      97308      6.945
1      1      Thomas 1996-03-31      06:53:00      61933      4.170
2      2      Maria 1993-04-23      11:17:00      130590      11.858
3      3      Jerry 2005-03-04      13:00:00      138705      9.340
4      4      Larry 1998-01-24      16:47:00      101004      1.389
..      ...      ...
995      995      Henry 2014-11-23      06:09:00      132483      16.655
996      996      Phillip 1984-01-31      06:30:00      42392      19.675
997      997      Russell 2013-05-20      12:39:00      96914      1.421
998      998      Larry 2013-04-20      16:45:00      60500      11.985
999      999      Albert 2012-05-15      18:24:00      129949      10.169

Senior Management      Team
0      1      Marketing
1      1      Marketing
2      0      Finance
3      1      Finance
4      1      Client Services
..      ...      ...
995      0      Distribution
996      0      Finance
997      0      Product
998      0      Business Development
999      1      Sales

```

[1000 rows x 8 columns]

```

Unnamed: 0 First Name Start Date Last Login Time Salary Bonus % \
0      0      Douglas 1993-08-06      12:42:00      97308      6.945
1      1      Thomas 1996-03-31      06:53:00      61933      4.170
2      2      Maria 1993-04-23      11:17:00      130590      11.858
3      3      Jerry 2005-03-04      13:00:00      138705      9.340
4      4      Larry 1998-01-24      16:47:00      101004      1.389
..      ...      ...
995      995      Henry 2014-11-23      06:09:00      132483      16.655
996      996      Phillip 1984-01-31      06:30:00      42392      19.675
997      997      Russell 2013-05-20      12:39:00      96914      1.421
998      998      Larry 2013-04-20      16:45:00      60500      11.985
999      999      Albert 2012-05-15      18:24:00      129949      10.169

Senior Management      Team
0      1      Marketing
1      1      Marketing
2      0      Finance
3      1      Finance
4      1      Client Services
..      ...      ...
995      0      Distribution
996      0      Finance
997      0      Product
998      0      Business Development
999      1      Sales

```

[1000 rows x 8 columns]

```
print("\nReplace values:\n")
n=pd.read_excel("E:/SHEIK/employees.xlsx")
print("Original data set:\n");
print(n)
y=n.replace({49:50})
print("\nUpdated dataset with replaced values: (49:50)\n")
print(y)
print("\nThe above updated data set will be stored in sample6.xlsx file....\n")
y.to_excel("E:/SHEIK/employees.xlsx")
```



Replace values:

Original data set:

	Unnamed: 0.1	Unnamed: 0	First Name	Start Date	Last Login Time	Salary	\
0	0	0	Douglas	1993-08-06	12:42:00	97308	
1	1	1	Thomas	1996-03-31	06:53:00	61933	
2	2	2	Maria	1993-04-23	11:17:00	130590	
3	3	3	Jerry	2005-03-04	13:00:00	138705	
4	4	4	Larry	1998-01-24	16:47:00	101004	
..
995	995	995	Henry	2014-11-23	06:09:00	132483	
996	996	996	Phillip	1984-01-31	06:30:00	42392	
997	997	997	Russell	2013-05-20	12:39:00	96914	
998	998	998	Larry	2013-04-20	16:45:00	60500	
999	999	999	Albert	2012-05-15	18:24:00	129949	

	Bonus %	Senior Management	Team
0	6.945	1	Marketing
1	4.170	1	Marketing
2	11.858	0	Finance
3	9.340	1	Finance
4	1.389	1	Client Services
..
995	16.655	0	Distribution
996	19.675	0	Finance
997	1.421	0	Product
998	11.985	0	Business Development
999	10.169	1	Sales

[1000 rows x 9 columns]

Updated dataset with replaced values: (49:50)

	Unnamed: 0.1	Unnamed: 0	First Name	Start Date	Last Login Time	Salary	\
0	0	0	Douglas	1993-08-06	12:42:00	97308	
1	1	1	Thomas	1996-03-31	06:53:00	61933	
2	2	2	Maria	1993-04-23	11:17:00	130590	
3	3	3	Jerry	2005-03-04	13:00:00	138705	
4	4	4	Larry	1998-01-24	16:47:00	101004	
..
995	995	995	Henry	2014-11-23	06:09:00	132483	
996	996	996	Phillip	1984-01-31	06:30:00	42392	
997	997	997	Russell	2013-05-20	12:39:00	96914	
998	998	998	Larry	2013-04-20	16:45:00	60500	
999	999	999	Albert	2012-05-15	18:24:00	129949	

	Bonus %	Senior Management	Team
0	6.945	1	Marketing
1	4.170	1	Marketing
2	11.858	0	Finance
3	9.340	1	Finance
4	1.389	1	Client Services
..
995	16.655	0	Distribution
996	19.675	0	Finance
997	1.421	0	Product

```
print("Original data set from Excel file:\n")
df=pd.read_excel("E:/SHEIK/employees.xlsx")
print(df)
print("\nExtract the particular record based on the isin() function condition:\n")
new=df['Start Date'].isin([49])
print(df[new])
```



Original data set from Excel file:

	Unnamed: 0.2	Unnamed: 0.1	Unnamed: 0	First Name	Start Date	\
0	0	0	0	Douglas	1993-08-06	
1	1	1	1	Thomas	1996-03-31	
2	2	2	2	Maria	1993-04-23	
3	3	3	3	Jerry	2005-03-04	
4	4	4	4	Larry	1998-01-24	

..
995	995	995	995	Henry	2014-11-23
996	996	996	996	Phillip	1984-01-31
997	997	997	997	Russell	2013-05-20
998	998	998	998	Larry	2013-04-20
999	999	999	999	Albert	2012-05-15

	Last Login Time	Salary	Bonus %	Senior Management	Team
0	12:42:00	97308	6.945	1	Marketing
1	06:53:00	61933	4.170	1	Marketing
2	11:17:00	130590	11.858	0	Finance
3	13:00:00	138705	9.340	1	Finance
4	16:47:00	101004	1.389	1	Client Services
..
995	06:09:00	132483	16.655	0	Distribution
996	06:30:00	42392	19.675	0	Finance
997	12:39:00	96914	1.421	0	Product
998	16:45:00	60500	11.985	0	Business Development
999	18:24:00	129949	10.169	1	Sales

[1000 rows x 10 columns]

Extract the particular record based on the isin() function condition:

Empty DataFrame
Columns: [Unnamed: 0.2, Unnamed: 0.1, Unnamed: 0, First Name, Start Date, Last Login Time, Salary, Bonus %, Senior Management, Team
Index: []

Start coding or [generate](#) with AI.

